2018-12-01

A computationally efficient formal method for discovering simutlaneous masking in medical alarms

Bolton, ML

10.1016/j.apacoust.2018.06.012
Applied Acoustics
Elsevier

A computationally efficient formal method for discovering simultaneous masking in medical alarms

Matthew L. Bolton[a,*], Judy Edworthy[b], Andrew D. Boyd[c], Jiajun Wei[a], Xi Zheng[a]

[a] *University at Buffalo, State University of New York, Department of Industrial and Systems Engineering, Buffalo, NT, USA*
[b] *Plymouth University, School of Psychology, Cognition Institute, Plymouth, UK*
[c] *University of Illinois at Chicago, Department of Biomedical and Health Information Sciences, Chicago, IL, USA*

Abstract

Numerous patient injuries and deaths have been caused by medical practitioners failing to respond to medical alarms. Simultaneous masking, where concurrently sounding medical alarms result in one or more being unhearable, is partially responsible for this problem. In previous work, we introduced a computational formal method capable of proving (formally verifying) if masking could occur in a modeled configuration of medical alarms. However, the scalability of the method limited the applicability and completeness of its analyses. In the work presented here, we show how we re-implemented the method to address these shortcomings. We evaluated the detection capabilities and scalability of the new version of the method with a series of realistic and synthetic case studies. Our results show that the new version of the method replicates and improves detection capabilities compared to the legacy method and does so with significant reductions in verification times. We discuss the patient safety implications of our results and explore directions for future research.

Keywords: Medical Alarms, Masking, Psychoacoustics, Formal Methods, Model Checking

1. Introduction

There are a number of problems with auditory medical alarms that can make them difficult to hear and respond to [13, 25]. According to the Pennsylvania Patient Safety Authority [24], there have been 194 documented problems with operators failing to properly respond to telemetry monitoring alerts between June 2004 and December 2008, including 12 deaths. According to a 2013 Sentinel Event Alert, 98 alarm-related non-response incidents were reported from January 2009 to June 2012. Eighty of these produced patient death, 13 resulted in a "permanent loss of function," and 5 caused patient hospital stays to be extended [57].

These problems are directly related to the fact that medical alarms sound at rates and in numbers that are incompatible with human sensory, perceptual, and cognitive capabilities [17, 25, 43, 57, 62]. For example, the Joint Commission [57] found that, in one day, hundreds of alarms can be produced by a single patient. This aggregates into tens of thousands of alarms sound daily across a given hospital. Because of these issues and the difficulties hospitals have had in solving them, the ECRI Institute has identified medical alarms as one of the most significant technological hazards to patient safety for more than a decade [23, 53].

Problems with the design of medical alarm auditory parameters are largely acknowledged as a contributor to these problems [23, 57, 58, 61]. In particular, the Joint Commission's 2014 National Patient Safety Goal (NPSG) to "improve the safety of clinical alarm systems" claimed that "individual alarm signals are difficult to detect" [58].

One problem that can make it difficult for humans to respond to medical alarms is simultaneous masking. In simultaneous masking, sounds playing in parallel can interact in ways that prevents humans from hearing one of or more of them due to limitations of the human sensory system [30]. A number of researchers have acknowledged that simultaneous masking is a problem with medical alarms and at least partially responsible for non-responses [26–28, 41, 46, 48, 49]. Furthermore, experimental results do indeed show that simultaneous masking exists in modern medical environments. Momtahan et al. [47], who analyzed 26 alarms from an operating room and 23 from an intensive care unit, found 25 pairs of alarms where one could be completely masked by the other. Toor et al. [59], discovered low priority sounds present in an operating room could easily mask higher priority alarms. It is important to note that these analyses only partially elucidate the problem because neither accounted for the additive effect of masking: where a sound can be masked by the interaction of multiple simultaneously playing sounds. Medical alarms (including those in the international standard [40]) are usually represented as melodies (patterns) of tonal sounds. These are particularly susceptible to simultaneous masking [12, 30]. Given that the probability of masking increases with the number of concurrently sounding alarms [12, 39, 65], the sheer number of alarms in modern medical environments [56] practically assures that masking is occurring.

Even with these results, the preponderance of medical alarm safety research has focused on other problems [25]. This is likely

*Corresponding author
Email address: mbolton@buffalo.edu (Matthew L. Bolton)

a symptom of the complexity of the masking problem. Specifically, it can be extremely difficult to detect auditory masking experimentally because it may only occur for particular interactions of multiple, concurrently-sounding medical alarms. Given the number of possible medical alarms, overlaps between them, and the masking potential associated with additive masking, it is practically impossible to evaluate every alarm configuration to find potential masking experimentally.

To address this problem, we developed a computational method [11, 34–36] that can detect masking in configurations of tonal medical alarms. The method uses a novel combination of psychoacoustics and model checking. The psychoacoustics describe simultaneous masking mathematically by relating sounds' frequency/tone and volume to the biologically-grounded masking effect the sounds have [2, 5, 12, 14, 15, 52]. Model checking is an automated approach for performing mathematical proofs (a process called formal verification) on models of concurrent systems [16]. When these technologies are used together in our method, an analyst can model the sounding behavior of multiple alarms and use model checking to prove whether the represented alarms can mask each other. This method has been used to analyze real medical alarm configurations [11, 36]. However, these analyses could take days to analyze even one alarm. Furthermore, the nature of the verification process limited the number of alarm interactions that could be considered in a proof. Thus, the analyses could conceivably miss interaction problems.

In the research presented here, we describe an improved version of our method. This improves its masking detection capabilities while simultaneously improving its scalability. Below we provide the necessary background to understand the different versions of our methods. We then present an updated version of the method and report results that demonstrate its improved scalability and analysis capabilities with both synthetic and realistic applications. We ultimately discuss the implications of our results and explore avenues of future research.

2. Background

Below, we review the relevant research on model checking, the psychoacoustics of simultaneous masking, and our method.

2.1. Model Checking

Model checking comes from the computer science field of "formal methods". In this context, formal methods are rigorous mathematical languages and techniques for specifying, modeling, and verifying systems [64]. Specifications describe desirable system properties, systems are modeled using mathematical languages, and verification mathematically proves whether or not the model satisfies the specification.

Model checking performs formal verification automatically [16]. A model describes a system's behavior, usually as a finite state machine: model variables with particular values represent state and changes in variable values (state) represent transitions. Specification properties are typically represented in a temporal logic [29], which use Boolean algebra, temporal operators, and system model variables to assert desirable system conditions. Verification processes prove whether the model satisfies the specification by exhaustively searching through the system model's statespace looking for violations. If the specification property proves to be true, the model checker returns a confirmation. If the property does not hold, the model checker returns an execution trace through the model called a counterexample. This shows exactly how the specification was violated. Model checking is especially good at discovering problems in systems with concurrency, where system elements can interact in ways unanticipated by designers and analysts [33]. Model checking is typically in the evaluation of discrete systems (where state is easily represented by discrete, categorical or ordinal variables). However, hybrid modeling and analysis techniques can account for continuous state variables [21, 38, 50]. They do this by mapping discrete model states (like the sounding state of an alarm) to continuous, real-valued quantities. For example, when using timed automata [1, 21], every model discrete state is assigned a time represented by a real number.

Model checking's major limitation is scalability. As concurrent elements are added to a formal model, the size of the model's statespace increases exponentially [16]. This "state explosion problem" can lead to situations where the model takes too long or is too big to verify. Because of this, analysts will often use abstraction techniques to model the systems they want to analyze [45].

Even with this limitation, model checking has demonstrated its utility for a variety of applications, especially for computer hardware and software [64]. Researchers have used model checking to successfully find and correct human factors issues in automated systems [6, 10, 20, 51, 63] and medical systems [3, 4, 7–9, 54, 60]. However, outside of our previous efforts on alarm masking modeling and detection[11, 34–36], no work has used model checking to find safety problems associated with human sensation and perception. Below we describe how our previous efforts worked. However, before we can do this, we need to explain the psychoacoustics of masking.

2.2. The Psychoacoustics of Simultaneous Masking

The psychoacoustics of simultaneous masking mathematically describe how the physical characteristics of a sound (its volume and tone/frequency) produce masking. These are based on the excitation patterns of the basilar membrane: the physical structure in the human ear that is predominately responsible for the human ability to distinguish between sounds [2, 5, 12, 14, 15, 52]. These models predict how a masking sound (the $masker$) will stimulate receptors on the inner ear's basilar membrane based on its volume and its relative frequency to a potentially masked sound (the $maskee$). This stimulation results in a higher volume threshold (in dB) that the volume of the $maskee$ must exceed to be perceivable [12].

The psychoacoustics of masking represent frequency on the Bark scale [22]. The Bark scale maps a frequency in Hz to a position on the basilar membrane (the spiral tube in the inner ear's cochlea) where that frequency most strongly stimulates the receptors (see Fig. 1). A sound's frequency in Hz (f_{sound}) is

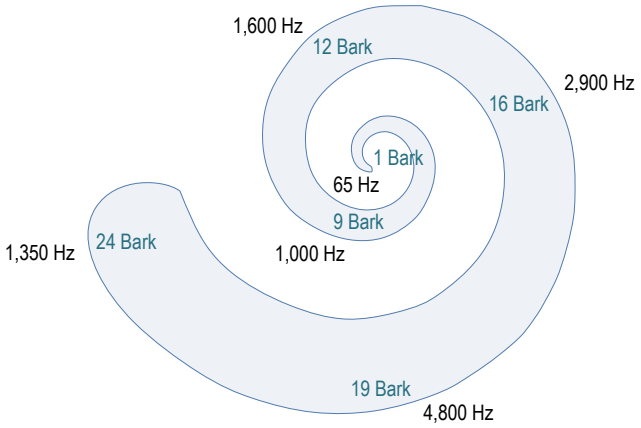


Figure 1: Depiction of how peak stimulation of sounds in Hz occurs at different Bark locations along the basilar membrane.

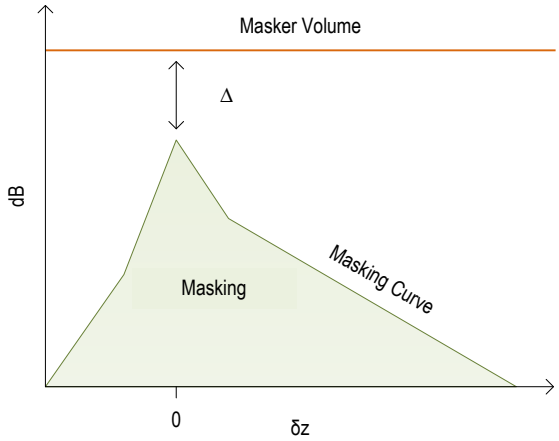


Figure 2: The basic shape of the masking curve [see (2)] with the MPEG [12] spreading function [see (4)]. Any sound whose frequency and volume falls below the masking curve will be masked.

converted to Barks by [22]

$$z_{sound} = 13 \cdot \arctan(0.00076 \cdot f_{sound}) + 3.5 \cdot \arctan\left((f_{sound}/7500)^2\right). \quad (1)$$

The "masking curve" then represents the masking threshold as:

$$\text{curve}_{masker}(z_{maskee}) = \text{spread}_{masker}(\delta z) + v_{masker} - \Delta. \quad (2)$$

v_{masker} is the volume of the masker in dB. δz is defined as

$$\delta z = z_{maskee} - z_{masker}, \quad (3)$$

where z_{maskee} and z_{masker} are the Bark scale frequency of the maskee and masker respectively. The spread_{masker} function models how the magnitude/volume of the masking threshold changes with respect to δz. Δ is the minimum difference between the masker's and maskee's volumes that can result in masking.

There are a number of psychoacoustic spreading functions for capturing the masking effect of different types of sounds [12]. Similarly, the formulation of Δ will depend on types of sounds being represented. In this research we use the following spreading function:

$$\text{spread}_{masker}(\delta z) = \begin{cases} -17 \cdot \delta z + 0.15 \cdot v_{masker} \cdot (\delta z - 1) \cdot \theta(\delta z - 1) & \text{for } \delta z \geq 0 \\ -(6 + 0.4 \cdot v_{masker}) \cdot |\delta z| - (11 + 0.4 \cdot v_{masker} \cdot (|\delta z| - 1)) \cdot \theta(|\delta z| - 1) & \text{otherwise} \end{cases} \quad (4)$$

where $\theta(x) = 1$ for $x \geq 0$ and $\theta(x) = 0$ otherwise. Δ was computed as

$$\Delta = 6.025 + 0.275 \cdot z_{masker} \text{ dB}. \quad (5)$$

These particular formulations were used for several reasons. First, they have been shown to be appropriate for modeling tonal sounds [2, 15]. They are also the basis of the MPEG audio codec [12] and have been well-validity. The shape of the masking curve (2) described by these parameters is shown in Fig. 2.

These psychoacoustics will indicate if a single sound can mask another. They were also the basis for previously published results [34, 35]. However, the combined masking threshold of multiple concurrent sounds can be greater than the sum of the masking effect of individual maskers. This effect is known as additive masking [12, 39]. This is modeled by adding up the masking curve values of each potential masker on the power scale. We can use the following to transform a volume (v in dB) onto the power scale

$$\text{power}\,(v) = 10^{v/10}. \quad (6)$$

Then, for a given potential maskee and N potential maskers, the additive masking threshold (in dB) is calculated as

$$\text{power}\,(\text{mthresh}_{maskee}) = \text{power}\,(\text{abs}_{maskee}) + \left(\sum_{n=1}^{N} \text{power}\,(\text{curve}_{masker_n}(z_{maskee}))^{\alpha}\right)^{1/\alpha}. \quad (7)$$

In the above, α is a positive constant [32]. abs_{maskee} is the absolute threshold of hearing (in dB) at the maskee's frequency (f_{maskee} in Hz). This is formulated as [55]

$$\text{abs}_{maskee} = 3.64 \cdot (f_{maskee}/1000)^{-0.8} - 6.5 \cdot e^{-0.6(f_{maskee}/1000 - 3.3)^2} + 10^{-3} \cdot (f_{maskee}/1000)^4. \quad (8)$$

These psychoacoustics have been used successfully to predict masking for normal human hearing for decades [12]. They were employed by researchers to identify when masking could occur sounds recorded in medical environments [59]. They were also the basis for lossy audio compression techniques, including those used in MPEG [12].

2.3. The Previous Version of Our Method

In the previous version of our method [11, 36] an analyst would follow the process shown in Fig. 3. In this, an analyst

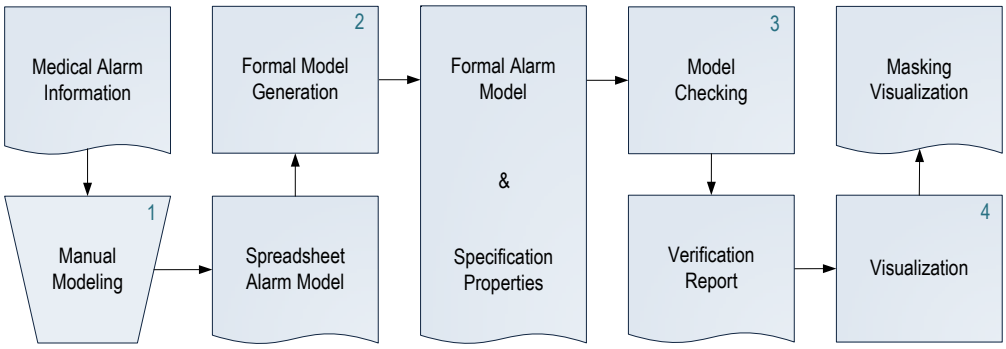


Figure 3: The flow model representation of our computational method for using model checking to discover simultaneous masking in configurations of medical alarms [36]. Numbers are used to show the order in which processes are performed. The formal modeling architecture used in the original method for the formal alarm model is show in Fig. 4. An example of a produced visualization is shown later in Fig. 8.

examined alarm documentation and described the behavior of the alarms using a MS Excel spreadsheet, where each alarm was represented as a sequence of tones (and pauses between tones) each with a defined frequency (Hz), volume (dB), and duration (s). When done describing alarms, the analyst used a computer program to automatically convert the alarm configuration into a formal model. This conversion also produced specifications for asserting the absence of masking. This included the lack of partial masking (no part of an alarm should ever be masked) and total masking (that no alarm should ever be masked for the entirety of its sounding cycle). Model checking was used to prove whether or not each of the generated specifications were true. In any situation where a counterexample was returned, the analyst could use a counterexample visualizer to identify where and how masking could occur.

The old version of the method used the architecture in Fig. 4 to represent the formal alarm configuration model. This was comprised of multiple, synchronously-composed sub-models. The clock sub-model used a timed automaton [1, 21] to advance model time (*globalTime*) and communicate it to the other sub-models. Each alarm was represented as a sub-model. Each could start or stop sounding at appropriate times and update its state based on its current state and how long it had been sounding. Alarm state represented each of the distinct tones or pauses that occurred over a complete sounding. A single masking computation sub-model used the current state of each alarm and the psychoacoustics of simultaneous masking to determine if any alarms were masked by the other sounding alarms. This sub-model also found the minimum of alarm next times (the *alarmNextTime* variables) to calculate a maximum time (*maxNextTime*) that the clock could be advanced to.

Because model checkers cannot handle the nonlinear arithmetic of model variables [18], our method used pre-computed functions (lookup tables) to capture nonlinear psychoacoustics. However, because the size of lookup tables can reduce the efficiency of a model (increase verification time), our old method minimized the number of necessary entries.

Critical to enabling this optimization is the concept of "power alpha," a value we introduced in [36]. By transforming a maskee's (any potentially masked alarm) volume and the masking effect of maskers into "power alpha" values using lookup tables, masking can be detected using only linear arithmetic operations. Figure 5 explains the formulation and rational for the "power alpha" values.

Our method used the relationship from Eq. (14) (Fig. 5) as the basis for its optimization. Specifically, a formal model generation process pre-computed each alarm's "power alpha" values when the alarm was both the potential maskee [using (12)] and masker [using (13)] for each of the alarm's states. These values were implemented in the formal model as lookup tables that were optimized to have the minimum number of entries. In the formal model, the masking computation sub-model treated each alarm as a potential maskee and all others as potential maskers. Thus, for a given potential maskee alarm, the masking computation sub-model would use the pre-computed lookup tables to perform the sum and comparison in (14) to determine if the potential maskee was indeed additively masked by the other sounding alarms. We used $\alpha = 0.33$, to capture the "over adding" of the masking effects of tones [44]. However, the method allowed for different analyst-specified α values.

To check a configuration of alarms modeled for masking using our original model, the analysts would need to check the generated properties (asserting the absence of any masking of a given alarm or the absence of total masking of the alarm) against the formal model using the infinite bounded model checker of the Symbolic Analysis Laboratory [18]. In doing this, the analyst would specify a search depth (a bound) on the total number of transitions considered in the analysis. Ideally this would be

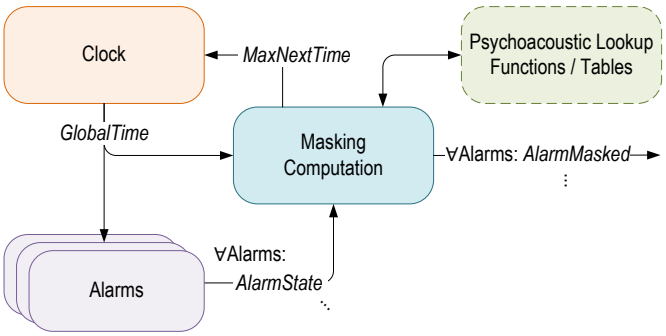


Figure 4: The architecture previously used for formally modeling a configuration of medical alarms in our method [36].

Note that the below use the equations defined in (1)–(8).

We know from the psychophysics of simultaneous additive masking [12] [see (7)] that a set of maskers will mask a maskee if

$$\text{power}\left(v_{maskee}\right) \leq \text{power}\left(\text{absolutethreshold}_{maskee}\right) + \left(\sum_{n=1}^{N} \text{power}\left(\text{curve}_{masker_n}\left(f_{maskee}\right)\right)^{\alpha}\right)^{1/\alpha}. \tag{9}$$

Using basic algebraic operations, we know that

$$\text{power}\left(v_{maskee}\right) - \text{power}\left(\text{absolutethreshold}_{maskee}\right) \leq \left(\sum_{n=1}^{N} \text{power}\left(\text{curve}_{masker_n}\left(f_{maskee}\right)\right)^{\alpha}\right)^{1/\alpha} \tag{10}$$

and thus that

$$\left(\text{power}\left(v_{maskee}\right) - \text{power}\left(\text{absolutethreshold}_{maskee}\right)\right)^{\alpha} \leq \sum_{n=1}^{N} \text{power}\left(\text{curve}_{masker_n}\left(f_{maskee}\right)\right)^{\alpha}. \tag{11}$$

If we let

$$\text{poweralpha}_{maskee} = \left(\text{power}\left(v_{maskee}\right) - \text{power}\left(\text{absolutethreshold}_{maskee}\right)\right)^{\alpha} \tag{12}$$

and

$$\text{maskingpoweralpha}_{masker}(maskee) = \text{power}\left(\text{curve}_{masker}\left(f_{maskee}\right)\right)^{\alpha}, \tag{13}$$

then we know that the maskee will be masked by the set of *N* maskers if

$$\text{poweralpha}_{maskee} \leq \sum_{n=1}^{N} \text{maskerpoweralpha}_{masker_n}(maskee). \tag{14}$$

Figure 5: Explanation of "power alpha" and how it can be used to determine if masking is occurring (adapted from [36]).

set, at minimum, to the total number of separate alarm events that could possibly occur in a given configuration. However, because increasing the search depth exponentially increased computational time [36], this was not always possible in practice.

This version of the method successfully improved upon the older version of the method [34, 35] by both being more usable (due to the spreadsheet-based modeling) and more scalable (the computational efficiency of using the optimized lookup tables) [11, 36]. In showing this, we analyzed the alarm system evaluated in the early versions of the method [34, 35] as well as the alarms from a real telemetry monitoring system, a GE CARESCAPE^{TM}Monitor B850 [31]. These analyses found a number of masking conditions. However, because of the search depth limitation of the method, it is possible some masking conditions were missed. Furthermore, even with the use of a less-than-optimal search depth, the analyses took prohibitively long to complete. For example, the analysis of partial masking for one alarm of the monitor took 4.57 days.

3. Objectives

In the work presented here, we show how we improve our computational technique for detecting simultaneous masking in configurations of medical alarms. In particular, we sought to improve its scalability and detection capabilities by rearchitecting how formal alarm models are constructed. This specifically worked by eliminating the need for the explicit formal modeling of a clock (see Fig. 4) by representing all relevant alarm times in a given model state. In addition to improving scalability, this eliminates the need for an analyst to specify a search depth when performing model checking. Thus, the exhaustiveness of our new approach was not limited by search depth, enabling a complete analysis of modeled alarm behavior. Below we describe how this new approach was realized. After this, we use the new approach to evaluate the alarm configurations reported in previous results [11, 34–36] to compare prediction performance and time. We also characterize how the method scales with a series of synthetic test cases. Finally, we interpret our results and explore their implications for future research.

4. The New Method

We updated our method to improve its scalability and remove the limitation that search depth placed on result completeness. Like the previous version, the new method uses the process shown in Fig. 3, where the analyst models alarms in an excel spreadsheet and automatically generates the model and specification properties used for checking for masking. However, in the new method, the formal alarm models generated use a new modeling architecture specifically designed to reduce the search depth required when model checking for masking. This was accomplished by ensuring that all of the information required for determine if masking was possible could manifest in the initial state of the model. This eliminated the need for a timed-automata based clock and thus ensured that search depths could no longer limit the detection capabilities of the method. This approach makes significant use of anonymous functions [19, 19, 37].

Anonymous functions come from the area of lambda calculus [37]. Specifically, anonymous functions are defined using lambda abstractions that describe functional mappings between types. In this work, we specify anonymous functions using lambda abstractions as defined in the language of of the symbolic analysis laboratory (SAL) [19]. This takes the form:

`LAMBDA` (*VariableDeclaration*) **:** *Expression*.

In this, the *VariableDeclaration* defines a variable of a specific type and the *Expression* defines how that type is transformed (which can be into the same or a different type) using the name of the variable. For example, let `TheArray` be an array of values, where allowable indices of the array are defined by the type `ARRAYINDEX`. With this, the lambda abstraction

```
LAMBDA (X : ARRAYINDEX) : TheArray[X] > 10
```

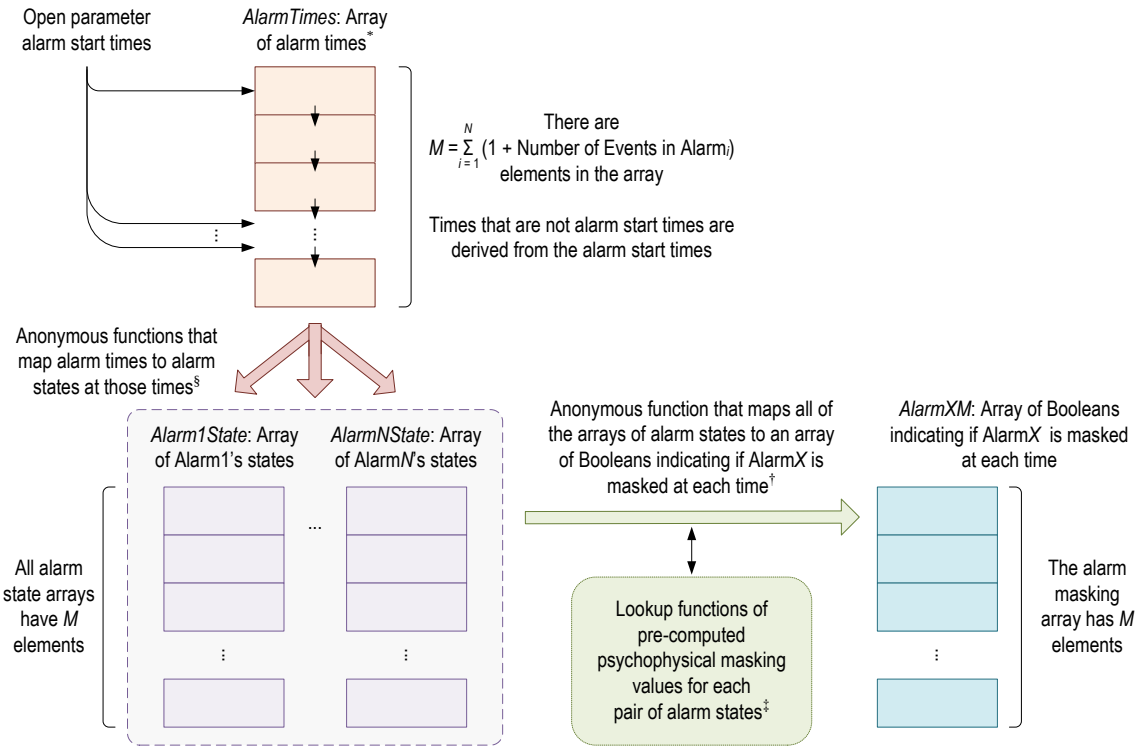


Figure 6: The new architecture for formally modeling a configuration of medical alarms in our method. Note that the superscript symbols $*$, §, †, and ‡ are used to concepts in this figure to corresponding concepts from Fig. 7.

describes an anonymous function that maps an array of integers to a sequence of Booleans where the mapping at a given X is true when the value at index X is greater than 10.

Anonymous functions are convenient for model checking because they give modelers a convenient notation for deriving complex systems concepts from a model's current state. This can enable a modeler to define complex concepts without having to perform computations using state transitions. In our new version of the method, we exploit anonymous functions to map an array of all of the important alarm time events for a given configuration (previously represented between model states using a timed automata; Fig. 4) to values indicating if an alarm is masked. Because all of the relevant alarm times are now represented in a models current state, verifications can be complete without the need for a model checker search depth greater than 0.

The architecture for our new approach is shown in Fig. 6. In this, all relevant analysis times are modeled in an array (*AlarmTimes*). Each entry in the array represents a time in which one alarm from the modeled configuration can change its state. The start times of each alarm are treated as open parameters (meaning they can be any possible time). The other event times for each alarm are then automatically computed based on when they sound relative to the previous alarm event (hence the arrows pointing between alarm elements in Fig. 6). Multiple anonymous functions are then used to compute the state of each alarm at each time. Specifically, the array of times is mapped into arrays of alarm states (*Alarm1State*...*AlarmNState*). There is one alarm state array for each of the *N* alarms in the modeled configuration. Each alarm state array has the same number of entries (*M*) as the array of alarm times, where each entry in an array represents the associated alarm's state at the time in the corresponding entry in the array of alarm times. An alarm's state is encapsulated by a unique name to indicate what tone is sounding or a generic non-state if the alarm is not sounding or in a pause.

A different, single anonymous function is then used to compute an array of Boolean values (*AlarmXM* with size *M*) to indicate if a given alarm (*AlarmX* in Fig. 6) is masked at each time based on the state of all of the alarms at each time. This functions makes use of the psychoacoustics of simultaneous masking to accomplish this. In particular, the function uses the same optimized lookup tables and the relationship from (14) (Fig. 5) as was used by the old version of the method. Note that if an alarm is not making any noise (it is not sounding or in a pause state; $Alarm0$) then this array indicates that it is masked. The distinction between masking that occurs when the alarm is not making noise and when it is making noise occurs when properties are being checked by the model checker.

To reduce complexity, a model based on the architecture in Fig. 6 is generated for each alarm in a configuration. This ensures that only psychoacoustic lookup tables are required for masking associated with a given alarm in a given model.

When a model based around the architecture in Fig. 6 is analyzed with the infinite bounded model checker, the analyst can check one of two generated specification properties. To check for total masking (that *AlarmX* can be rendered completely unhearable by other sounding alarms), the analyst checks a property of the form

$$\mathbf{G}\neg(\forall\, i \leq M : AlarmXM[i]). \tag{15}$$

This asserts that for all paths through the model (**G**) it should never be true ($\neg$) that for all the times (i), $AlarmX$ is masked ($Masking[i]$). Note that here $AlarmX$ represents the alarm that is being treated as the maskee. To check for partial masking (that the alarm can be masked in part by other sounding alarms), the analyst checks a property of the form:

$$\mathbf{G}\left(\begin{array}{r} \forall\, i \leq M : AlarmXM[i] \\ \Rightarrow AlarmXState = Alarm0 \end{array} \right). \quad (16)$$

This asserts that for all possible paths (**G**) and all of the times (i) through the model, if $AlarmX$ is masked then this implies that the alarm is not making any noise ($AlarmXState = Alarm0$; the alarm is either not sounding or in a pause between alarm sounds). When checking both of these properties, because all of the possible alarm sounding patterns can be present in one of the infinite model initial states, search depth can be set to 0.

We modified the software implementation of the method (Fig. 3) to support the new modeling architecture (Fig. 6) and specification property patterns from (15) and (16). Like the previous version [36], the updated implementation generated models and specifications for use with the infinite bounded model checker of SAL [18] However, unlike the previous version, this new implementation generated multiple formal models, one for each included alarm. This allowed us to realize the totality of the new optimized approach by only having to generate lookup tables for values associated with treating a given alarm as the maskee. This meant that an analyst would need to run verifications using different formal models instead of just one. From a practitioner's perspective, this is a trivial difference. Figure 7 shows how the formal model and specifications for a given alarm (Alarm*X*) are generically formulated using the input language of SAL [19].

5. Testing and Results

We evaluated our new method using two different types of tests. In the first, we evaluated the same cases originally reported in [36] to determine whether the new version of the method could detect the same masking conditions as the old one and to compare computational efficiency (verification time) of the two approaches. In the second, we set out to fully characterize the scalability of the new method using a series of synthetic case studies based on the parameters of the international medical alarm standard [40]. Both are described below.

Note that all of these analyses followed the procedure specified by our method (Fig. 3). First, the analyzed alarms were modeled using the reported parameters in an excel spreadsheet. Second, we used our updated software to automatically generate formal models and specification properties for evaluating the masking potential of each alarm. Third, SAL's infinite bounded model checker was used to formally check whether each alarm was could be partially or totally masked. All of these verification results reported below were conducted on a computer workstation with a 3.3 GHz Intel Xeon processor and 64 GB of RAM running Linux Mint. Finally, our method's visualization was used (where appropriate) to plot the results.

Table 1: Case Study 1 Alarm Configuration (adapted from [36])

Name	Freq. (Hz)	Vol. (dB)	Time (s)
Alarm 1	261	80	0.250
	0	0	0.100
	370	80	0.250
Alarm 2	277	60	0.150
	0	0	0.050
	277	60	0.150
Alarm 3	524	85	0.200
	0	0	0.075
	294	85	0.200

Note. Alarm tones are listed vertically (from top to bottom) based on the order that they sound in a given alarm's cycle. A pause is indicated by a volume or frequency of 0.

5.1. Reproduction of Previous Analyses

We originally evaluated three case studies [36]. Below we discuss how each of these cases was evaluated with the new method and compare the new results to those obtained in [36]. Each case study provides different challenges. Case study 1 is a simple example where masking can manifest between pairs of alarms without additive masking. Case study 2 is a more complex application where additive masking is required to detect masking conditions. Case study 3 is a realistic application: an actual telemetry monitoring system.

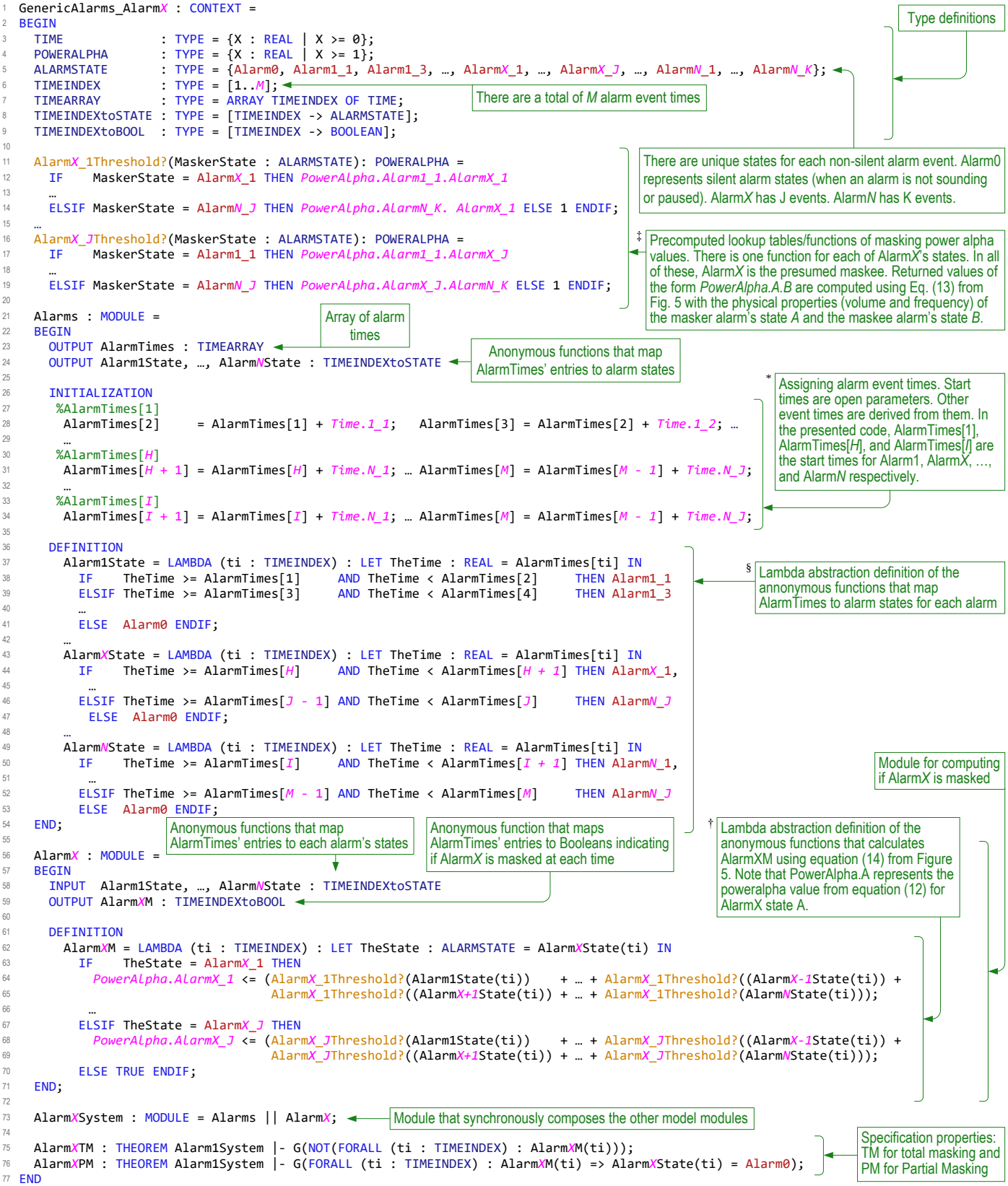
5.1.1. Case Study 1: The Original Application

In the case study originally presented in [34] and reevaluated in [36], there were three alarms (Table 1). All of these alarms had two tones separated by a pause. The frequencies, tone and pause lengths, and volumes were consistent with those commonly found in medical alarms [40, 47].

With old method [36], analyses were conducted on four different configurations: one for configurations with each possible pair of alarms from Table 1 and one with all three alarms. Each configuration was also modeled and evaluated with the new method. A comparison of the analysis results with both methods is reported in Table 2. These show that the same outcome was achieved between the two methods when each specification property was checked. The produced counterexamples showed that the same masking conditions were discovered using both methods.

The results in Table 2 also demonstrate the scalability improvements of our new method. The new method was able to perform each of the analyses significantly faster than the older method, where reductions in verification times varied from 66.67% to 99.98%.

This case study illustrates the improvements in verification time achieved by the new method while preserving the detection capabilities. However, this case study does not evaluate the additive masking detection capabilities of the method. This is because multiple overlapping alarms were not required to produce the discovered masking conditions. Additive masking detection is evaluated in the next case study.




```
1  GenericAlarms_AlarmX : CONTEXT =
2  BEGIN
3    TIME             : TYPE = {X : REAL | X >= 0};
4    POWERALPHA       : TYPE = {X : REAL | X >= 1};
5    ALARMSTATE       : TYPE = {Alarm0, Alarm1_1, Alarm1_3, …, AlarmX_1, …, AlarmX_J, …, AlarmN_1, …, AlarmN_K};
6    TIMEINDEX        : TYPE = [1..M];
7    TIMEARRAY        : TYPE = ARRAY TIMEINDEX OF TIME;
8    TIMEINDEXtoSTATE : TYPE = [TIMEINDEX -> ALARMSTATE];
9    TIMEINDEXtoBOOL  : TYPE = [TIMEINDEX -> BOOLEAN];
10
11   AlarmX_1Threshold?(MaskerState : ALARMSTATE): POWERALPHA =
12     IF     MaskerState = AlarmX_1 THEN PowerAlpha.Alarm1_1.AlarmX_1
13     …
14     ELSIF MaskerState = AlarmN_J THEN PowerAlpha.AlarmN_K. AlarmX_1 ELSE 1 ENDIF;
15   …
16   AlarmX_JThreshold?(MaskerState : ALARMSTATE): POWERALPHA =
17     IF     MaskerState = Alarm1_1 THEN PowerAlpha.Alarm1_1.AlarmX_J
18     …
19     ELSIF MaskerState = AlarmN_J THEN PowerAlpha.AlarmX_J.AlarmN_K ELSE 1 ENDIF;
20
21   Alarms : MODULE =
22   BEGIN
23     OUTPUT AlarmTimes : TIMEARRAY
24     OUTPUT Alarm1State, …, AlarmNState : TIMEINDEXtoSTATE
25
26     INITIALIZATION
27      %AlarmTimes[1]
28       AlarmTimes[2]     = AlarmTimes[1] + Time.1_1;   AlarmTimes[3] = AlarmTimes[2] + Time.1_2; …
29       …
30      %AlarmTimes[H]
31       AlarmTimes[H + 1] = AlarmTimes[H] + Time.N_1; … AlarmTimes[M] = AlarmTimes[M - 1] + Time.N_J;
32       …
33      %AlarmTimes[I]
34       AlarmTimes[I + 1] = AlarmTimes[I] + Time.N_1; … AlarmTimes[M] = AlarmTimes[M - 1] + Time.N_J;
35
36     DEFINITION
37       Alarm1State = LAMBDA (ti : TIMEINDEX) : LET TheTime : REAL = AlarmTimes[ti] IN
38         IF     TheTime >= AlarmTimes[1]     AND TheTime < AlarmTimes[2]     THEN Alarm1_1
39         ELSIF TheTime >= AlarmTimes[3]     AND TheTime < AlarmTimes[4]     THEN Alarm1_3
40         …
41         ELSE  Alarm0 ENDIF;
42       …
43       AlarmXState = LAMBDA (ti : TIMEINDEX) : LET TheTime : REAL = AlarmTimes[ti] IN
44         IF     TheTime >= AlarmTimes[H]     AND TheTime < AlarmTimes[H + 1] THEN AlarmX_1,
45           …
46         ELSIF TheTime >= AlarmTimes[J - 1] AND TheTime < AlarmTimes[J]     THEN AlarmN_J
47          ELSE  Alarm0 ENDIF;
48       …
49       AlarmNState = LAMBDA (ti : TIMEINDEX) : LET TheTime : REAL = AlarmTimes[ti] IN
50         IF     TheTime >= AlarmTimes[I]     AND TheTime < AlarmTimes[I + 1] THEN AlarmN_1,
51           …
52         ELSIF TheTime >= AlarmTimes[M - 1] AND TheTime < AlarmTimes[M]     THEN AlarmN_J
53         ELSE  Alarm0 ENDIF;
54   END;
55
56   AlarmX : MODULE =
57   BEGIN
58     INPUT  Alarm1State, …, AlarmNState : TIMEINDEXtoSTATE
59     OUTPUT AlarmXM : TIMEINDEXtoBOOL
60
61     DEFINITION
62       AlarmXM = LAMBDA (ti : TIMEINDEX) : LET TheState : ALARMSTATE = AlarmXState(ti) IN
63         IF     TheState = AlarmX_1 THEN
64           PowerAlpha.AlarmX_1 <= (AlarmX_1Threshold?(Alarm1State(ti))    + … + AlarmX_1Threshold?((AlarmX-1State(ti)) +
65                                   AlarmX_1Threshold?((AlarmX+1State(ti)) + … + AlarmX_1Threshold?(AlarmNState(ti)));
66           …
67         ELSIF TheState = AlarmX_J THEN
68           PowerAlpha.AlarmX_J <= (AlarmX_JThreshold?(Alarm1State(ti))    + … + AlarmX_JThreshold?((AlarmX-1State(ti)) +
69                                   AlarmX_JThreshold?((AlarmX+1State(ti)) + … + AlarmX_JThreshold?(AlarmNState(ti)));
70         ELSE TRUE ENDIF;
71   END;
72
73   AlarmXSystem : MODULE = Alarms || AlarmX;
74
75   AlarmXTM : THEOREM Alarm1System |- G(NOT(FORALL (ti : TIMEINDEX) : AlarmXM(ti)));
76   AlarmXPM : THEOREM Alarm1System |- G(FORALL (ti : TIMEINDEX) : AlarmXM(ti) => AlarmXState(ti) = Alarm0);
77 END
```

Figure 7: Generic model code for implementing the architecture from Fig. 6 for checking if an arbitrary alarm (Alarm$X$) is masked within a configuration of N alarms. This uses the input language of SAL [19]. Lines that start with % indicate code comments. Magenta variables represent values inserted into the code during generation. Ellipses are used to show where incremental code is produced based on the presented patterns. Annotations (in green boxes) are used to describe the code. Where applicable, variables in this are named consistently with their corresponding concepts from Fig. 6. This includes *AlarmTimes*, M, N, *Alarm1State–AlarmNState*, and *AlarmXM*. The assignment of alarm times (Fig. 6*) is shown in lines 27–34. The lambda abstractions that define the anonymous functions that map alarm times to alarm states (Fig. 6§) are shown in lines 37–53. The lambda abstraction that defines the anonymous function that maps all of the alarm states to the masking-indicating Boolean array for Alarm$X$ (Fig. 6†) is in 62–70. The psychoacoustic mapping functions used for this (Fig. 6‡) are defined in lines 11–19.

5.1.2. Case Study 2: Additive Masking Detection

The second case study (originally from [36]) evaluated the alarms in Table 3. These were chosen because they, with the exception of tone timing, are similar to medium priority sounds from the international medical alarm standard's reserved sounds [40] and we wanted to test whether our method could reproduce the additive masking capabilities of the previous method.

As in the previous analyses [36], we used these alarms to construct four different configurations: one for each possible pair of alarms and one with all three. By using our method to evaluate all four configurations, we were able to determine if our method could reproduce the additive masking results. Specifically, if we found masking that occurred due to two or more alarms overlapping a maskee, where masking did not occur when each potential masker alarm overlapped the maskee by itself, then our method could find additive masking conditions.

Table 2: Case Study 1 Verification Results

Model			Original Method		New Method		%
Alarms	Alarm	Spec.	Time (s)	Result	Time (s)	Result	Decrease
1 & 2	1	Partial	0.15	✓	0.03	✓	80.00%
		Total	0.11	✓	0.03	✓	72.73%
	2	Partial	0.47	×	0.02	×	95.74%
		Total	0.24	✓	0.02	✓	91.67%
1 & 3	1	Partial	0.11	✓	0.02	✓	81.82%
		Total	0.12	✓	0.03	✓	75.00%
	3	Partial	0.16	✓	0.02	✓	87.50%
		Total	0.10	✓	0.02	✓	80.00%
2 & 3	2	Partial	1.24	×	0.03	×	97.58%
		Total	0.17	✓	0.02	✓	88.24%
	3	Partial	0.15	✓	0.03	✓	80.00%
		Total	0.09	✓	0.03	✓	66.67%
1, 2, & 3	1	Partial	6.65	✓	0.06	✓	99.10%
		Total	1.58	✓	0.04	✓	97.47%
	2	Partial	89.97	×	0.07	×	99.92%
		Total	148.29	×	0.04	×	99.97%
	3	Partial	3.74	✓	0.06	✓	98.40%
		Total	1.46	✓	0.04	✓	97.26%

Note. ✓ indicates a verification confirmation and × indicates a verification failure with a counterexample. Time represents the total verification time in seconds. % Decrease is computed as 100% · (Original Time - New Time) / Original Time. In all verification results, the number of visited states is not reported because this is not calculated by SAL's infinite bounded model checker.

Table 3: Case Study 2 Alarm Configuration (adapted from [36])

Name	Freq. (Hz)	Vol. (dB)	Time (s)
Alarm A	261	84	0.1
	0	0	0.1
	329	84	0.1
	0	0	0.1
	392	84	0.1
Alarm B	261	84	0.1
	0	0	0.1
	329	84	0.1
	0	0	0.1
	293	84	0.1
Alarm C	523	84	0.1
	0	0	0.1
	293	84	0.1
	0	0	0.1
	392	84	0.1

Table 4: Case Study 2 Verification Results

Model			Original Method		New Method		%
Alarms	Alarm	Spec	Time (s)	Result	Time (s)	Result	Decrease
A & B	Alarm A	Partial	4.48	✓	0.03	✓	99.33%
		Total	1.09	✓	0.02	✓	98.17%
	Alarm B	Partial	3.42	✓	0.04	✓	98.83%
		Total	2.34	✓	0.03	✓	98.72%
A & C	Alarm A	Partial	4.31	✓	0.04	✓	99.07%
		Total	0.99	✓	0.03	✓	96.97%
	Alarm C	Partial	2.89	✓	0.04	✓	98.62%
		Total	1.85	✓	0.04	✓	97.84%
B & C	Alarm B	Partial	3.96	✓	0.04	✓	98.99%
		Total	1.40	✓	0.03	✓	97.86%
	Alarm C	Partial	3.60	✓	0.03	✓	99.17%
		Total	1.48	✓	0.04	✓	97.30%
A , B & C	Alarm A	Partial	189.2	✓	0.24	✓	99.87%
		Total	13.56	✓	0.12	✓	99.12%
	Alarm B	Partial	670.23	×	0.27	×	99.96%
		Total	9.83	✓	0.15	✓	98.47%
	Alarm C	Partial	815.29	×	0.34	×	99.96%
		Total	16.94	✓	0.12	✓	99.29%

We checked the specifications for each alarm (for both partial and total masking) using both versions of the method. For the original, for models containing two alarms, verification search depths were set to 12. A search depth of 18 was used for models with three alarms. Search depths of 0 were used for all analyses with the new method. Results are reported in Table 4.

As with the previous case study, these results show that the new method can replicate the result of the previous version while offering significant improvements in verification time (96.97% – 99.96% decreases with reductions factors of 33 – 2,482.33). These results are significant because they further confirm that our new method is capable of detecting additive masking. The counterexamples for these analyses show that partial masking of the third tone of Alarm B occurs when it sounds at the same time as the first tone from Alarm A and the second tone of Alarm C. The second tone of Alarm C can be partially masked when sounding concurrently with the first tone of Alarm A and the third tone of Alarm B. Because no masking occurred in the models with only two alarms, the masking observed in the three-alarm model is additive.

5.1.3. Case Study 3: The GE CARESCAPE™ Telemetry Monitor

To evaluate a realistic application, we used our new method to analyze the alarms in the GE CARESCAPE™ Monitor B850 [31], a telemetry monitoring system compatible with the international medical alarm standard [40] (an analysis originally reported in [36]). The GE monitor had the alarms described in Table 5. There were four high-priority alarms that played identical ten-tone alarm melodies (including the same timings) at different volumes, a medium-priority alarm with three tones in its melody, and a one tone low-priority alarm. The analysis allowed any of the included alarms to sound concurrently.

We modeled the alarms from Table 5 in both versions of the method. Each alarm was evaluated with both methods to determine if it was ever partially or totally masked. Because

Table 5: Alarms from Case Study 3, the GE CARESCAPE Telemetry Monitoring System

Name	Freq. (Hz)	Vol. (dB)	Time (s)	Name	Freq. (Hz)	Vol. (dB)	Time (s)	Name	Freq. (Hz)	Vol. (dB)	Time (s)
CPU-C1	523	72	0.1	D15K	523	81	0.1	D19KT	523	82	0.1
	0	0	0.1		0	0	0.1		0	0	0.1
	698	72	0.1		698	81	0.1		698	82	0.1
	0	0	0.1		0	0	0.1		0	0	0.1
	784	72	0.1		784	81	0.1		784	82	0.1
	0	0	0.3		0	0	0.3		0	0	0.3
	880	72	0.1		880	81	0.1		880	82	0.1
	0	0	0.1		0	0	0.1		0	0	0.1
	988	72	0.1		988	81	0.1		988	82	0.1
	0	0	1.0		0	0	1.0		0	0	1
	523	72	0.1		523	81	0.1		523	82	0.1
	0	0	0.1		0	0	0.1		0	0	0.1
	698	72	0.1		698	81	0.1		698	82	0.1
	0	0	0.1		0	0	0.1		0	0	0.1
	784	72	0.1		784	81	0.1		784	82	0.1
	0	0	0.3		0	0	0.3		0	0	0.3
	880	72	0.1		880	81	0.1		880	82	0.1
	0	0	0.1		0	0	0.1		0	0	0.1
	988	72	0.1		988	81	0.1		988	82	0.1
	0	0	5.0		0	0	5.0		0	0	5
SystemHigh	523	84	0.1	SystemMedium	523	83	0.2	SystemLow	523	79	0.2
	0	0	0.1		0	0	0.2				
	698	84	0.1		784	83	0.2				
	0	0	0.1		0	0	0.2				
	784	84	0.1		988	83	0.2				
	0	0	0.3		0	0	19.0				
	880	84	0.1								
	0	0	0.1								
	988	84	0.1								
	0	0	1.0								
	523	84	0.1								
	0	0	0.1								
	698	84	0.1								
	0	0	0.1								
	784	84	0.1								
	0	0	0.3								
	880	84	0.1								
	0	0	0.1								
	988	84	0.1								
	0	0	5.0								

Note. CPU-C1, D15K, D19KT, and SystemHigh are high-priority alarms. SystemMedium is a medium-priority alarm. SystemLow is a low priority alarm.

of the complexity of the model, we anticipated that the original method would have scalability problems. Thus, in the results reported in [36] (and reproduced here) we attempted to minimize verification search depths. Specifically, all properties were verified iteratively starting with the minimum depth capable of detecting masking. If no masking was found, the search depth was increased by one for each verification until masking was discovered or the verification took a prohibitively long time. For partial masking, this meant search depths started at 2 and increased from there. For total masking, search depths started at the total number of states in the associated alarm and were iteratively increased up to 21. Search depths greater than 21 were not considered because of the amount of time required for the analyses. Because a depth of 21 would encapsulate what was likely to be the worst possible masking condition for the three high-priority alarms (when they all sounded at the same time as each other due to them all having the same tones), this was seen as sufficient. Verifications done with the new method were performed with a search depth of 0.

Table 6: Case Study 3 Verification Results

		Original Method			New Method		
Alarm	Spec	Depth	Time (s)	Result	Depth	Time (s)	Result
CPU-C1	Partial	2	145.70	×	0	55,265.73	×
	Total	21	60,967.05	×	0	2,361.15	×
D15K	Partial	2	135.21	×	0	70,236.22	×
	Total	21	145,870.50	✓	0	302.14	×
D19KT	Partial	2	135.21	×	0	75,160.80	×
	Total	21	148,252.81	✓	0	413.19	×
SystemHigh	Partial	2	139.02	×	0	80,352.56	×
	Total	21	395,441.48	✓	0	73.31	×
SystemMedium	Partial	2	104.24	×	0	45,424.01	×
	Total	21	203,702.73	✓	0	84.32	✓
SystemLow	Partial	2	81.24	×	0	992.29	×
	Total	4	216.66	×	0	76.51	×

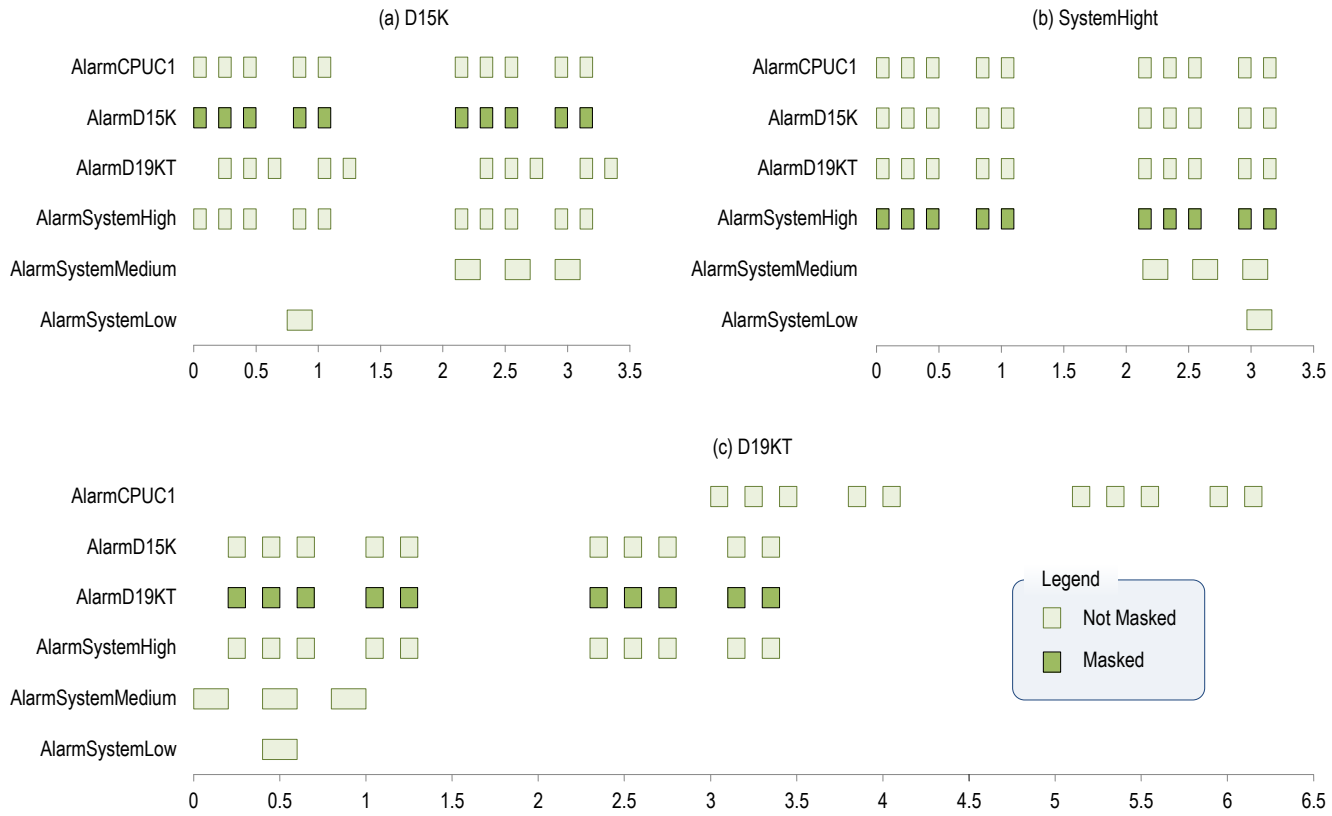


Figure 8: Plots illustrating the new total masking conditions found using the new masking verification method: (a) D15K, (b) SystemHigh, and (c) D19KT. In these, boxes indicate when an alarm is making noise in accordance with the sounds shown in Table 5.

Results are shown in Table 6. In these, the new method was able to reproduce the partial masking results for the alarms of the GE CARESCAPE. Specifically, partial masking was observed for all of the alarms. The new method also found the same total masking conditions reported in [36]. However, the new method also found additional total masking conditions not found with the original method. Specifically, the new method found that D15K, D19KT, and SystemHigh could also be totally masked. While unexpected, these results actually make sense given that the search depth was limited in the original analyses due to scalability limitations of the original method. The method's visualization of the total masking of these three alarms (Fig. 8) shows that this probably occurred because more than 21 alarm events (the maximum search depth in the original analyses) were required to achieve these masking results.

Finally, it is important to note that, unlike the results from the previous analyses, the partial masking verification results actually took longer with the new method than the original one (Table 6). This occurs because, in the original analyses, partial masking was found at search depths of 2. Thus, the original method was considering significantly fewer alarm interaction conditions. For the total masking analyses (where significantly larger search depths were originally used), the new method reduced verification times from between 64.69% and 99.98%. This, coupled with the additional masking conditions discovered with the new method, clearly demonstrates that the new method is both more complete than the old approach (the analysis is no longer limited by search depth and thus offers genuine proof of masking conditions) and more computationally efficient.

5.2. Scalability Analyses

The following analyses were constructed to characterize how our method scales. To do these, we developed an artificial case study which had three base alarms (Table 7). In this configuration, Alarm 1 was designed to never be masked and Alarm 2 was designed to be both partially and totally masked by Alarm 3. Each of these alarms contains ten tones followed by a pause, making them have the maximum number of alarm events allowed in IEC 60601-1-8. In the scalability analyses, variations of this configuration were created, where both the number of alarm events and number of alarms were varied. Specifically, cases were created where all of the alarms in the test case had between 1 and 10 tones (all followed by a pause) where the tones included in the analyses started with the first tones for the alarms in Table 7 and iteratively adding each of their tones (and their following pause) as the number of tones increased. For each specific number of tones, separate cases were created which contained between 3 and 8 alarms (this was chosen as the upper limit because there are 8 reserved alarm sounds in IEC 60601-1-8). For each case where there were more than 3 alarms, Alarm 3 was reproduced to add the additional alarms.

For all of these cases, the partial masking and total masking of both Alarm 1 and Alarm 2 were recorded to get metrics (verification times) for both discovering the presence and ab-

Table 7: Alarms Used in the Scalability Analyses

Name	Freq (Hz)	Vol (dB)	Time (s)	Name	Freq (Hz)	Vol (dB)	Time (s)	Name	Freq (Hz)	Vol (dB)	Time (s)	
Alarm 1	150	60	0.1	Alarm 2	1000	84	0.1	Alarm 3	150	84	0.1	...
	0	0	0.1		0	0	0.1		0	0	0.1	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	0.1		0	0	0.1		0	0	0.1	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	0.3		0	0	0.3		0	0	0.3	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	0.1		0	0	0.1		0	0	0.1	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	1.0		0	0	1.0		0	0	1.0	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	0.1		0	0	0.1		0	0	0.1	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	0.1		0	0	0.1		0	0	0.1	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	0.3		0	0	0.3		0	0	0.3	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	0.1		0	0	0.1		0	0	0.1	
	150	60	0.1		1000	84	0.1		150	84	0.1	
	0	0	5.0		0	0	5.0		0	0	5.0	

Note. In analyses with more than three alarms, each additional alarm is a duplicate of Alarm 3.

sence of partial and total masking. In the results, the method always found that Alarm 1 was totally or partially masked and that Alarm 2 was never masked (as expected). Verification times can be found in Table 8. In all of these results, verification times appeared to increase exponentially with both the number of alarms and the number of tones in the alarms. To check this, we fit exponential functions to the data using linear regression and computed the coefficient of determination (R^2). In all cases, $R^2 > 0.9$ indicating a very good fit of an exponential function to the data. This confirmed our observation.

Even though verification time increases exponentially with the number of tones and the number of alarms in the model, the scalability results are encouraging. Specifically, all of the verifications were able to complete, with the longest taking just under three days.

It is also important to note that the analyses that indicated the presence of total masking (Table 8) took significantly less time to complete than the other analyses. This is not surprising given that analyses finish as soon as they find a property violation. It is also convenient for analyses, because it means that problems can be discovered and potentially correct quickly.

6. Discussion

This work has introduced an approach that significantly improves the scalability and completeness of a formal-methods-based approach for discovering when masking can cause medical alarms to be imperceptible. This development was achieved by allow the full sounding cycle of alarm configurations to be considered in the set of model initial states.

The results presented for the legacy analyses (compared to analyses originally reported in [36]) demonstrate that the method is capable of achieving the same level of masking detection seen with the previous version of the method. This is shown by the fact that the new method was able to find the positive masking results for case studies 1, 2, and 3. Due to the scalability limitations of the original method, the original verification analyses of case study 3 did not use a search depth sufficient enough to consider all possible model interactions. Thus, the analyses of case study 3 with the new method found masking conditions not previously discovered with the original method. This demonstrates the improved detection capabilities afforded by the completeness of the new method. This is a significant result because it shows that the new method will not miss critical alarm interactions.

Not only was the new method complete, but it also significantly improved scalability. In particular, the analyses for case studies 1 and 2 saw across the board reductions in verification times ranging from 66.67% to 99.97% for comparable analyses. Similar reductions were observed for the total masking analyses with case study 3. Conversely, increases in verification times were observed for partial masking. However, this is due to the improved completeness of the new method. Thus, these results do not constitute a serious problem for the new method.

The method improvements are responsible for the performance observed in the synthetic case studies for assessing scalability. In particular, the analysis that took the longest time to complete (total masking of alarm 1 with 10 tones and 8 alarms) completed in less than three days (249,676.89 s; Table 8). These results have important implications for the use of the method.

The results for case study 3 and the scalability results show that the method can be realistically used by designers to evaluate the masking potential of alarms from modern medical devices. Specifically, for a complex configuration with up to 8 alarms each with up to 10 tones (and 10 pauses), it will take less than three days of computational time to run an evaluation on each alarm. Furthermore, analyses of different alarms can be run in parallel. Thus a full design could be evaluated in three days with enough conveintional computational resources. It is worth noting that 8 alarms (with ten tones each) is a fair number of alarms to consider in a given analyses when evaluating a design. This is

Table 8: Scalability Analyses Verification Times (in seconds)

#	# Alarms						
Tones	3	4	5	6	7	8	R^2
Partial Masking of Alarm 1							
1	0.03	0.05	0.15	0.55	1.63	4.29	0.99
2	0.05	0.21	0.60	2.55	12.61	34.96	~ 1
3	0.14	0.58	1.87	8.06	64.44	215.03	0.99
4	0.27	1.18	3.08	36.72	177.60	793.33	0.99
5	0.39	2.07	15.59	89.63	596.68	3,241.92	~ 1
6	0.63	3.68	30.71	244.53	1,919.88	10,587.44	~ 1
7	1.04	6.15	46.10	335.16	4,916.96	25,279.72	~ 1
8	1.34	6.91	97.56	1,587.23	11,283.61	53,273.24	0.99
9	1.70	16.05	82.43	2,178.08	20,215.13	133,548.58	0.99
10	3.05	9.85	239.35	2,354.85	31,542.81	213,370.02	0.99
R^2	0.97	0.91	0.95	0.96	0.97	0.97	
Total Masking of Alarm 1							
1	0.02	0.05	0.12	0.50	1.42	3.85	~ 1
2	0.05	0.17	0.65	3.09	9.39	33.20	~ 1
3	0.13	0.38	2.19	13.61	59.33	229.04	~ 1
4	0.20	1.03	5.84	44.19	225.64	1,064.94	~ 1
5	0.35	1.63	11.96	103.41	665.59	3,753.30	~ 1
6	0.56	2.64	23.19	214.12	2,373.25	13,000.95	~ 1
7	0.85	5.39	38.07	519.12	4,505.68	37,017.86	~ 1
8	1.19	6.45	57.90	1,049.41	12,825.34	69,355.22	0.99
9	1.60	8.13	103.84	1,986.15	25,995.18	148,159.88	0.99
10	2.07	14.75	163.11	4,366.77	48,322.69	249,676.89	0.99
R^2	0.96	0.95	0.95	0.97	0.97	0.97	
Partial Masking of Alarm 2							
1	0.03	0.04	0.12	0.39	1.16	4.78	0.98
2	0.05	0.22	1.05	4.49	16.72	51.12	~ 1
3	0.16	0.77	6.58	27.01	97.77	330.36	0.99
4	0.31	2.43	26.63	121.44	537.13	1,460.08	0.98
5	0.55	4.64	71.59	404.97	1,463.16	5,349.96	0.98
6	1.05	16.79	214.23	1,216.66	4,786.87	18,030.59	0.97
7	1.30	32.31	345.01	2,476.75	8,357.33	42,150.85	0.97
8	2.57	67.48	890.27	4,239.88	23,432.29	69,819.05	0.96
9	3.76	164.43	1,398.34	8,187.31	32,677.39	111,553.38	0.96
10	5.86	204.45	1,909.07	16,608.25	48,411.44	189,249.12	0.96
R^2	0.98	0.97	0.94	0.94	0.93	0.93	
Total Masking of Alarm 2							
1	0.03	0.03	0.06	0.10	0.14	0.20	0.97
2	0.04	0.11	0.22	0.49	0.66	1.26	0.98
3	0.10	0.30	0.54	1.08	1.92	3.28	0.98
4	0.19	0.51	1.17	2.54	4.84	7.45	0.98
5	0.33	1.01	2.42	4.76	9.01	14.68	0.98
6	0.49	1.55	4.02	8.62	16.01	29.16	0.98
7	0.79	2.48	6.57	13.64	25.20	53.91	0.99
8	1.02	3.55	9.75	20.94	41.22	76.28	0.98
9	1.71	5.50	14.66	30.80	58.38	104.28	0.98
10	2.28	7.75	21.28	42.01	86.12	160.37	0.98
R^2	0.98	0.95	0.96	0.95	0.94	0.94	

made clear by considering the fact that a device as complex as the telemetry monitoring system evaluated for case study 3 only had 6 alarms. Even the IEC 60601-1-8 international medical alarm standard only contains 8 reserved alarm sounds, which can have up to 10 tones. Thus our method is fully capable of evaluating the alarms of conventional designed standards as well as the reserved sounds of IEC 60601-1-8. As such, the work presented here has the potential to allow designers to reduce the likelihood that alarms in their devices will be masked. This should improve the probability that medical practitioners will hear the alarms, respond to them appropriately, and thus avoid adverse health outcomes. We plan to publish a free implementation of our analysis method so that designers will be able to use it in future device designs.

It is important to note that our method does not consider the likelihood or risk of any particular masking condition. We do not view this as a major limitation of our work because any alarm masking could be fatal in a medical environment. Furthermore, predicting the probability that any particular alarm will sound at a given time will likely be difficult and have high variance. However, should such estimates become possible, we could adapt our method for use with emerging probabilistic model checking techniques [42]. Probabilistic model checking is similar to more convention model checking approaches, but allows probabilities and consequence to be associated with different formal model transition. This enables its use for predicting the probability and risk of outcomes. Future work should investigate how our method could be used with probabilistic model checking to enable such capabilities.

The results of the telemetry monitoring systems (case 3) are particularly troubling because there are ways for all of the high priority alarms in the system to be masked. This system was not chosen for analysis because we anticipated it having masking problems. On the contrary, it was the only medical device we could obtain detailed alarm information for. This provides further evidence that the masking problem is more serious than previously thought and worthy of consideration in system design.

The results for the telemetry monitoring system are even more concerning given that its alarms were designed in compliance with IEC 60601-1-8, the international medical alarm standard [40]. This suggests that masking is a critical issue in the standard. This conclusion is further bolstered by the fact that the standard was the inspiration for the sounds analyzed under case study 2. Thus, in future work we plan to use our method to evaluate the IEC 60601-1-8 international standard. In particular, IEC 60601-1-8 contains a number of reserved sounds: standard alarm melodies for representing common medical alarm concepts. In total, there are eight reserved alarm sounds where, depending on the priority, alarms could have 1, 3, or 10 tones separated by a pause. As such, the scalability analyses presented here show that our method is capable of handling the complexity of the alarms in the international standard.

It is worth noting that the international medical alarm standard specifies that alarms have additional (lower volume) frequencies in each tone. This can make IEC 60601-1-8 compliant alarms harmonically richer. The alarms evaluated in this paper only considered the primary frequency of alarms. However, our current version of the method is able to account for this feature of IEC 60601-1-8 and the nature of our new architecture allows the masking effect of additional frequencies to be considered without additional impact on scalability. This is due to the fact that the effect of additional frequencies can be incorporated into the pre-computed "power alpha" values that our method uses.

Thus, should our analysis of IEC 60601-1-8 prove insightful, this work has the potential to make recommendations for improving the international medical alarm standard. This could significantly decrease the likelihood that medical alarms are masked. In a medical environment, where seconds can mean the difference between life and death, this could have profound implications for patient safety and health.

7. Acknowledgement

Research reported in this paper was supported by the Agency for Healthcare Research and Quality under award number R18HS024679. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Agency for Healthcare Research and Quality.

References

[1] Alur, R., & Dill, D. L. (1994). A theory of timed automata. *Theoretical computer science*, *126*, 183–235.

[2] Ambikairajah, E., Davis, A., & Wong, W. (1997). Auditory masking and MPEG-1 audio compression. *Electronics & Communication Engineering Journal*, *9*, 165–175.

[3] Baksi, D. (2008). Formal interaction specification in public health surveillance systems using π-calculus. *Computer methods and programs in biomedicine*, *92*, 115–120.

[4] Baksi, D. (2009). Model checking of healthcare domain models. *Computer Methods and Programs in Biomedicine*, *96*, 217–225.

[5] Baumgarte, F., Ferekidis, C., & Fuchs, H. (1995). A nonlinear psychoacoustic model applied to ISO/MPEG layer 3 coder. In *Proceedings of the Audio Engineering Society Convention*. New York: Audio Engineering Society.

[6] Bolton, M. L. (2017). Novel developments in formal methods for human factors engineering. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 715–717). Los Angeles: Sage volume 61.

[7] Bolton, M. L., & Bass, E. J. (2009). A method for the formal verification of human interactive systems. In *Proceedings of the 53rd Annual Meeting of the Human Factors and Ergonomics Society* (pp. 764–768). Santa Monica: HFES.

[8] Bolton, M. L., & Bass, E. J. (2010). Formally verifying human-automation interaction as part of a system model: Limitations and tradeoffs. *Innovations in Systems and Software Engineering: A NASA Journal*, *6*, 219–231.

[9] Bolton, M. L., Bass, E. J., & Siminiceanu, R. I. (2012). Generating phenotypical erroneous human behavior to evaluate human-automation interaction using model checking. *International Journal of Human-Computer Studies*, *70*, 888–906.

[10] Bolton, M. L., Bass, E. J., & Siminiceanu, R. I. (2013). Using formal verification to evaluate human-automation interaction in safety critical systems, a review. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, *43*, 488–503.

[11] Bolton, M. L., Hasanain, B., Boyde, A. D., & Edworthy, J. (2016). Using model checking to detect masking in IEC 60601-1-8-compliant alarm configurations. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (pp. 636–640). Los Angeles: SAGE Publications.

[12] Bosi, M., & Goldberg, R. E. (2003). *Introduction to Digital Audio Coding and Standards*. New York: Springer.

[13] Boyd, A. D. (2010). *Centralized Telemetry Monitoring Center Human Factors Report*. Technical Report University of Illinois at Chicago.

[14] Brandenburg, K., & Bosi, M. (1997). Overview of MPEG audio: Current and future standards for low bit-rate audio coding. *Journal of the Audio Engineering Society*, *45*, 4–21.

[15] Brandenburg, K., & Stoll, G. (1994). ISO/MPEG-1 audio: A generic standard for coding of high-quality digital audio. *Journal of the Audio Engineering Society*, *42*, 780–792.

[16] Clarke, E. M., Grumberg, O., & Peled, D. A. (1999). *Model checking*. Cambridge: MIT Press.

[17] Cvach, M. (2012). Monitor alarm fatigue: An integrative review. *Biomedical Instrumentation & Technology*, *46*, 268–277.

[18] De Moura, L., Owre, S., Rueß, H., Rushby, J., Shankar, N., Sorea, M., & Tiwari, A. (2004). SAL 2. In *Proceedings of the 16th International Conference on Computer Aided Verification* (pp. 496–500). Springer.

[19] de Moura, L., Owre, S., & Shankar, N. (2003). *The SAL language manual*. Technical Report CSL-01-01 Computer Science Laboratory, SRI International Menlo Park.

[20] Dix, A. J. (1991). *Formal methods for interactive systems* volume 16. London: Academic Press.

[21] Dutertre, B., & Sorea, M. (2004). *Timed systems in SAL*. Technical Report NASA/CR-2002-211858 SRI International.

[22] E. Zwicker and R. Feldtkeller (1967). *Das Ohr als Nachrichtenempfnger*. Stuttgart: Hirzel Verlag.

[23] ECRI Institute (2014). Top 10 health technology hazards for 2015. *Health Devices*, *November*. URL: `http://www.ecri.org/2015hazards`.

[24] ECRI Institute, & ISMP (2009). Connecting remote cardiac monitoring issues with care areas. *Pennsylvania Patient Safety Authority*, *6*, 79–83. URL: `http://patientsafetyauthority.org/ADVISORIES/AdvisoryLibrary/2009/Sep6(3)/Pages/79.aspx`.

[25] Edworthy, J. (2013). Medical audible alarms: A review. *Journal of the American Medical Informatics Association*, *20*, 584–589.

[26] Edworthy, J., & Hellier, E. (2005). Fewer but better auditory alarms will improve patient safety. *Quality and Safety in Health Care*, *14*, 212–215.

[27] Edworthy, J., & Hellier, E. (2006). Alarms and human behaviour: Implications for medical alarms. *British Journal of Anaesthesia*, *97*, 12–17.

[28] Edworthy, J., & Meredith, C. S. (1994). Cognitive psychology and the design of alarm sounds. *Medical Engineering & Physics*, *16*, 445–449.

[29] Emerson, E. A. (1990). Temporal and modal logic. In J. van Leeuwen, A. R. Meyer, M. Nivat, M. Paterson, & D. Perrin (Eds.), *Handbook of Theoretical Computer Science* chapter 16. (pp. 995–1072). Cambridge: MIT Press.

[30] Fastl, H., & Zwicker, E. (2006). *Psychoacoustics: Facts and models* volume 22. Springer.

[31] GE Healthcare (2010). *CARESCAPE^{TM}Monitor B850 Technical Specifications Supplement*. Technical Report 2040386-084D General Electric Company.

[32] Green, D. M. (1967). Additivity of masking. *The Journal of the Acoustical Society of America*, *41*, 1517–1525.

[33] Grumberg, O., & Veith, H. (2008). *25 Years of Model Checking: History, Achievements, Perspectives*. Berlin: Springer.

[34] Hasanain, B., Boyd, A., & Bolton, M. (2016). Using model checking to detect simultaneous masking in medical alarms. *IEEE Transactions on Human-Machine Systems*, *46*, 174–185.

[35] Hasanain, B., Boyd, A., & Bolton, M. L. (2014). An approach to model checking the perceptual interactions of medical alarms. In *Proceedings of the 2014 International Annual Meeting of the Human Factors and Ergonomics Society* (pp. 822–826). Santa Monica: HFES.

[36] Hasanain, B., Boyd, A. D., Edworthy, J., & Bolton, M. L. (2017). A formal approach to discovering simultaneous additive masking between auditory medical alarms. *Applied Ergonomics*, *58*, 500–514.

[37] Henk, B. (1984). The lambda calculus: Its syntax and semantics. *Studies in logic and the foundations of Mathematics*, .

[38] Henzinger, T. A. (1996). The theory of hybrid automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science* (pp. 278–292). Washington: IEEE Computer Society.

[39] Humes, L. E., & Jesteadt, W. (1989). Models of the additivity of masking. *The Journal of the Acoustical Society of America*, *85*, 1285–1294.

[40] IEC 60601-1-8 (2003-08-14). *Medical Electrical Equipment - Part 1-8*. Geneva: International Electrotechnical Commission.

[41] Konkani, A., Oakley, B., & Bauld, T. J. (2012). Reducing hospital noise: A review of medical device alarm management. *Biomedical Instrumentation & Technology*, *46*, 478–487.

[42] Kwiatkowska, M., Norman, G., & Parker, D. (2011). Prism 4.0: Verification of probabilistic real-time systems. In *International conference on computer aided verification* (pp. 585–591). Springer.

[43] Lacherez, P., Seah, E., & Sanderson, P. (2007). Overlapping melodic alarms are almost indiscriminable. *Human Factors*, *49*, 637–645.

[44] Lutfi, R. A. (1983). Additivity of simultaneous masking. *The Journal of the Acoustical Society of America*, *73*, 262–267.

[45] Mansouri-Samani, M., Pasareanu, C. S., Penix, J. J., Mehlitz, P. C., OMalley, O., Visser, W. C., Brat, G. P., Markosian, L. Z., & Pressburger, T. T. (2007). *Program Model Checking: A Practitioners Guide*. Technical Report Intelligent Systems Division, NASA Ames Research Center Moffett Field.

[46] Meredith, C., & Edworthy, J. (1995). Are there too many alarms in the intensive care unit? An overview of the problems. *Journal of Advanced Nursing*, *21*, 15–20.

[47] Momtahan, K., Hetu, R., & Tansley, B. (1993). Audibility and identification of auditory alarms in the operating room and intensive care unit. *Ergonomics*, *36*, 1159–1176.

[48] Patterson, R. D. (1982). *Guidelines for Auditory Warning Systems on Civil Aircraft*. Civil Aviation Authority.

[49] Patterson, R. D., Mayfield, T. F., Patterson, R. D., & Mayfield, T. F. (1990). Auditory warning sounds in the work environment. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, *327*, 485–492.

[50] Podelski, A., & Wagner, S. (2006). Model checking of hybrid systems: From reachability towards stability. In *Hybrid Systems: Computation and Control* (pp. 507–521). Springer.

[51] Rukšėnas, R., Back, J., Curzon, P., & Blandford, A. (2008). Formal modelling of salience and cognitive load. In *Proceedings of the 2nd International Workshop on Formal Methods for Interactive Systems* (pp. 57–75). Amsterdam: Elsevier Science Publishers.

[52] Schroeder, M. R., Atal, B. S., & Hall, J. (1979). Optimizing digital speech coders by exploiting masking properties of the human ear. *The Journal of the Acoustical Society of America*, *66*, 1647–1652.

[53] Stead, W. W., & Lin, H. S. (Eds.) (2009). *Computational Technology for Effective Health Care: Immediate Steps and Strategic Directions*. Atlanta: National Academies Press.

[54] Ten Teije, A., Marcos, M., Balser, M., van Croonenborg, J., Duelli, C., van Harmelen, F., Lucas, P., Miksch, S., Reif, W., Rosenbrand, K. et al. (2006). Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, *36*, 193–209.

[55] Terhardt, E. (1979). Calculating virtual pitch. *Hearing Research*, *1*, 155–182.

[56] Thangavelu, S. D., Ifeachor, E., Edworthy, J., Yunus, J., & Chinna, K. (2014). Challenges and recommendation of clinical alarm system in intensive care units from user perspective. In *2014 IEEE Region 10 Symposium* (pp. 366–369). Piscataway: IEEE.

[57] The Joint Commission (2013). Medical device alarm safety in hospitals. *Sentinel Even Alert*, *50*.

[58] The Joint Commission (2013). Npsg.06.01.01: Improve the safety of clinical alarm systems. *Joint Commission Perspectives*, *33*.

[59] Toor, O., Ryan, T., & Richard, M. (2008). Auditory masking potential of common operating room sounds: A psychoacoustic analysis. In *Anesthesiology* (p. A1207). Park Ridge: American Society of Anesthesiologists volume 109.

[60] van Breda, W., Hoogendoorn, M., Eiben, A., & Berking, M. (2017). Assessment of temporal predictive models for health care using a formal method. *Computers in Biology and Medicine*, *87*, 347–357.

[61] Vockley, M. (2014). *Clinical Alarm Management Compendium*. Arlington: AAMI Foundation.

[62] Way, R. B., Beer, S. A., & Wilson, S. J. (2014). Whats that noise? Bedside monitoring in the emergency department. *International Emergency Nursing*, *22*, 197–201.

[63] Weyers, B., Bowen, J., Dix, A., & Palanque, P. (Eds.) (2017). *The Handbook of Formal Methods in Human-Computer Interaction*. Berlin: Springer.

[64] Wing, J. M. (1990). A specifier's introduction to formal methods. *Computer*, *23*, 8, 10–22, 24.

[65] You, Y. (2010). *Audio Coding: Theory and Applications*. Springer Science & Business Media.